Allen-Bradley

PowerFlex® Communications

EtherNet/IP Adapter

22-COMM-E
FRN 1.xxx

User Manual

Rockwell Automation

Using Explicit Messaging

Chapter 6 provides information and examples that explain how to use Explicit Messaging to configure and monitor the EtherNet/IP adapter installed and connected to the PowerFlex 40 drive.

Topic	Page
About Explicit Messaging	6-1
Formatting Explicit Messages	6-2
Performing Explicit Messages	6-4

Topic	Page
About the Example Explicit Messages	6-5
Example Get Attribute Single Message	6-6
Example Set Attribute Single Message	6-8

ATTENTION: Risk of injury or equipment damage exists. The examples in this publication are intended solely for purposes of example. There are many variables and requirements with any application. Rockwell Automation, Inc. does not assume responsibility or liability (to include intellectual property liability) for actual use of the examples shown in this publication.

ATTENTION: Risk of equipment damage exists. If Explicit Messages are programmed to write parameter data to Non-Volatile Storage (NVS) frequently, the NVS will quickly exceed its life cycle and cause the drive to malfunction. Do not create a program that frequently uses Explicit Messages to write parameter data to NVS.

Refer to Chapter 5 for information about the I/O image, using Logic Command/Status and Reference/Feedback.

About Explicit Messaging

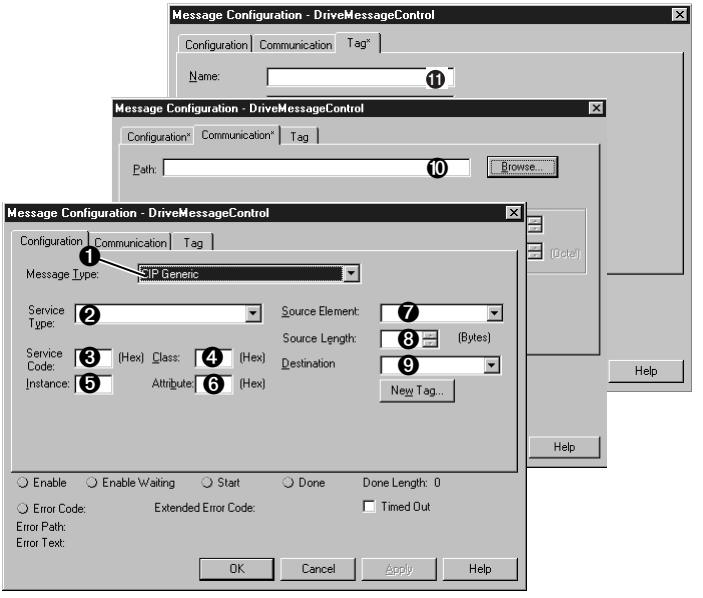
Explicit Messaging is used to transfer data that does not require continuous updates. With Explicit Messaging, you can configure and monitor a slave device's parameters on the EtherNet/IP network.

Formatting Explicit Messages

Explicit Messages for the ControlLogix Controller

ControlLogix scanners and bridges accommodate both downloading Explicit Message Requests and uploading Explicit Message Responses. The scanner or bridge module can accommodate one request or response for each transaction block. Each transaction block must be formatted as shown in Figure 6.1.

Figure 6.1 ControlLogix Message Format in RSLogix 5000



Refer to page 6-3 for a description of the data that is required in each box (1 – 11).

TIP: To display the Message Configuration dialog box in RSLogix 5000, add a message instruction (MSG), create a new tag for the message (properties: Base tag type, MESSAGE data type, controller scope), and click the blue box inside the message.

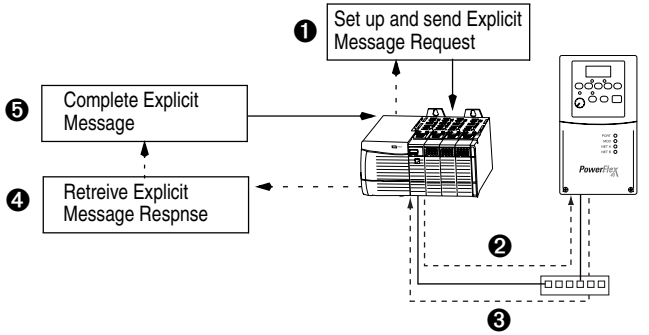
ControlLogix Message Requests and Responses

Box	Description
❶	**Message Type** The message type must be CIP Generic.
❷	**Service Type** The service type indicates the service (for example, Get Attribute Single or Set Attribute Single) that you want to perform. Available services depend on the class and instance that you are using. Refer to Appendix C, EtherNet/IP Objects.
❸	**Service Code** The service code is the code for the requested EtherNet/IP service. This value changes based on the Service Type that has been selected. In most cases, this is a read-only box. If you select “Custom” in the Service Type box, then you need to specify a service code in this box (for example, 4B for a Get Attributes Scattered service or 4C for a Set Attributes Scattered service).
❹	**Class** The class is an EtherNet/IP class. Refer to Appendix C, EtherNet/IP Objects, for available classes.
❺	**Instance** The instance is an instance (or object) of an EtherNet/IP class. Refer to Appendix C, EtherNet/IP Objects, for available instances for each class.
❻	**Attribute** The attribute is a class or instance attribute. Refer to Appendix C, EtherNet/IP Objects, for available attributes for each class or instance.
❼	**Source Element** This box contains the name of the tag for any service data to be sent from the scanner or bridge to the adapter and drive.
❽	**Source Length** This box contains the number of bytes of service data to be sent in the message.
❾	**Destination** This box contains the name of the tag that will receive service response data from the adapter and drive.
❿	**Path** The path is the route that the message will follow. **Tip:** Click **Browse** to find the path or type in the name of an adapter that you previously mapped.
⓫	**Name** The name for the message.

Performing Explicit Messages

There are five basic events in the Explicit Messaging process. The details of each step will vary depending on the controller. Refer to the documentation for your controller.

Figure 6.2 Explicit Message Process



Event

1. You format the required data and set up the ladder logic program to send an Explicit Message request to the scanner or bridge module (download).

2. The scanner or bridge module transmits the Explicit Message Request to the slave device over the Ethernet network.

3. The slave device transmits the Explicit Message Response back to the scanner. The data is stored in the scanner buffer.

4. The controller retrieves the Explicit Message Response from the scanner's buffer (upload).

5. The Explicit Message is complete.

About the Example Explicit Messages

These examples show how to format and execute the following types of Explicit Messages using a ControlLogix controller:

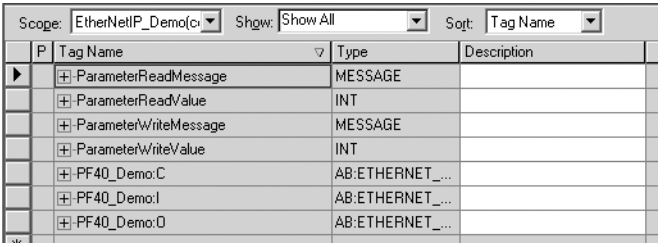
- Get Attribute Single
- Set Attribute Single

Message Formats

When formatting an example message, refer to Formatting Explicit Messages in this chapter for an explanation of the content of each box.

Also, to format and execute these example messages, you need the Controller tags displayed in Figure 6.3.

Figure 6.3 Controller Tags for Explicit Messages



Ladder Logic Rungs

The ladder logic rungs for the examples in this chapter can be appended after rung 12 in the ladder logic program (Figure 5.5) in Chapter 5, Using I/O Messaging.

Source and Destination Data

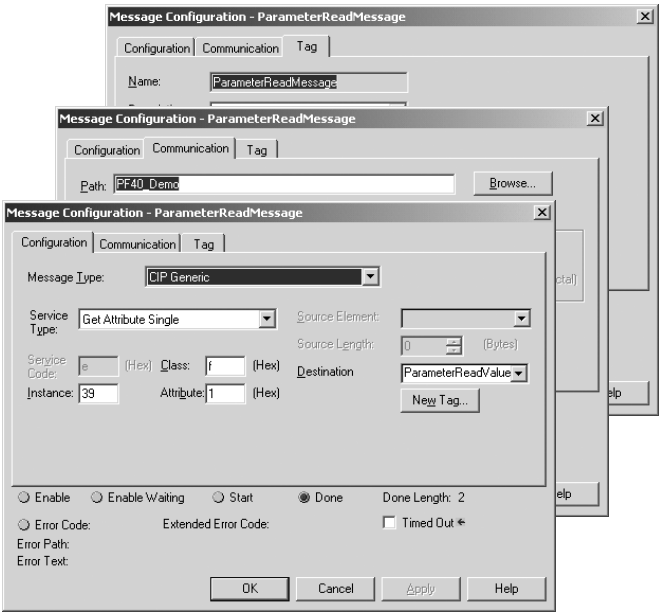
The example values for the source and destination data that appear in this chapter may vary in your application.

Example Get Attribute Single Message

A Get Attribute Single message reads a single attribute value. In this example, we read the value of a parameter in a PowerFlex 40 drive.

Example Message Format

Figure 6.4 Message Format for a Get Attribute Single Message



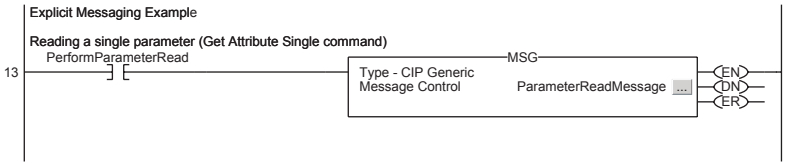
The following table identifies key settings for the message format:

Configuration	Value	Description	Refer to . . .
Service Type (1)	Get Attribute Single	Read parameter data	C-27
Service Code (1)	e (Hex.)	Get_Attribute_Single	C-12
Class	f (Hex.)	DSI Parameter Object	C-24
Instance	39 (Dec.)	Parameter 39 - [Accel Time 1]	C-24
Attribute	1 (Hex.)	Parameter Value	C-25
Destination	ParameterReadValue	Controller tag for response data	—

(1) The default setting for Service Type is “Custom,” which enables you to enter a Service Code that is not available from the Service Type pulldown menu. When you select a Service Type other than “Custom” from the pulldown menu, an appropriate Hex. value is automatically assigned to the Service Code box which grays out (unavailable).

Example Ladder Logic Rung

Figure 6.5 Example Get Attribute Single Message



Example Destination Data

In this example, the Get Attribute Single message reads **Parameter 39 - [Accel Time 1]** in the PowerFlex 40 drive and returns its value to the destination tag named ParameterReadValue.

Figure 6.6 Example Destination Data from a Get Attribute Single Message

	ParameterReadValue	100		Decimal	INT

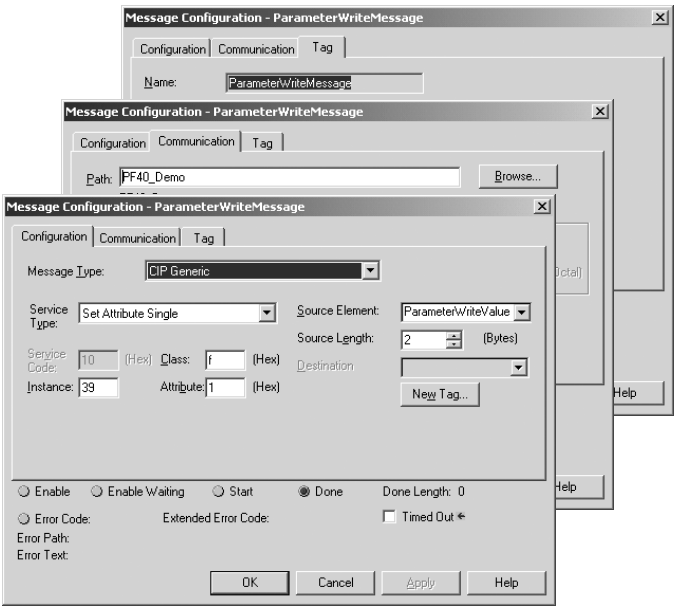
The acceleration time is 10.0 seconds.

Example Set Attribute Single Message

A Set Attribute Single message writes a value for a single attribute. In this example, we write the value of a parameter in a PowerFlex 40 drive.

Example Message Format

Figure 6.7 Message Format for a Set Attribute Single Message



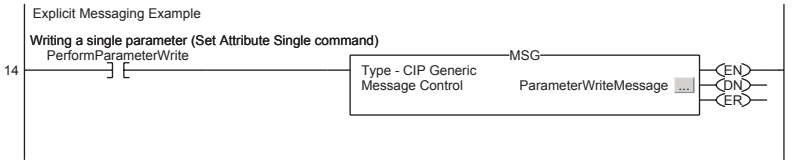
The following table identifies key settings for the data format:

Configuration	Value	Description	Refer to . . .
Service Type (1)	Set Attribute Single	Write parameter data	C-27
Service Code (1)	10 (Hex.)	Set_Attribute_Single	C-12
Class	f (Hex.)	DSI Parameter Object	C-24
Instance	39 (Dec.)	Parameter 39 - [Accel Time 1]	C-24
Attribute	1 (Hex.)	Parameter Value	C-25
Source Element	ParameterWriteValue	Controller tag for write data	—
Source Length	2 bytes	One 16-bit word of data is sent	—

(1) The default setting for Service Type is “Custom,” which enables you to enter a Service Code that is not available from the Service Type pulldown menu. When you select a Service Type other than “Custom” from the pulldown menu, an appropriate Hex. value is automatically assigned to the Service Code box which grays out (unavailable).

Example Ladder Logic Rung

Figure 6.8 Example Set Attribute Single Message



Example Source Data

In this example, the Set Attribute Single message writes 100, the value in the source tag named ParameterWriteValue, to **Parameter 39 - [Accel Time 1]** in the PowerFlex 40 drive.

Figure 6.9 Example Source Data from a Set Attribute Single Message

	ParameterWriteValue	100		Decimal	INT

10.0 seconds is written to the parameter.

Notes:

EtherNet/IP Objects

Appendix C provides information about the EtherNet/IP objects that can be accessed using Explicit Messages. For information on the format of Explicit Messages and example ladder logic programs, refer to Chapter 6, Using Explicit Messaging.

Object	Class Code Hex.	Class Code Dec.	Page
Identity Object	0x01	1	C-2
Assembly Object	0x04	4	C-4
Register Object	0x07	7	C-6
Parameter Object	0x0F	15	C-9
Parameter Group Object	0x10	16	C-13
PCCC Object	0x67	103	C-15

Object	Class Code Hex.	Class Code Dec.	Page
DPI Device Object	0x92	146	C-21
DPI Parameter Object	0x93	147	C-24
DPI Fault Object	0x97	151	C-28
DPI Diagnostic Object	0x99	153	C-30
TCP/IP Interface Object	0xF5	245	C-32
Ethernet Link Object	0xF6	246	C-34

TIP: Refer to the EtherNet/IP specification for more information about EtherNet/IP objects. Information about the EtherNet/IP specification is available on the ODVA web site (http://www.odva.org).

Supported Data Types

Data Type	Description
BYTE	8-bit unsigned integer
WORD	16-bit unsigned integer
DWORD	32-bit unsigned integer
LWORD	64-bit unsigned integer
SINT	8-bit signed integer
USINT	8-bit unsigned integer
INT	16-bit signed integer
UINT	16-bit unsigned integer
DINT	32-bit signed integer
UDINT	32-bit unsigned integer
BOOL	8-bit value -- low bit is true or false
BOOL[n]	Array of n bits
STRING[n]	Array of n characters
SHORT_STRING	1-byte length indicator + that many characters
STRUCT	Structure name only - no size in addition to elements
CONTAINER	32-bit parameter value - sign extended if necessary
TCHAR	8 or 16-bit character
REAL	32-bit floating point

Identity Object

Class Code

Hexadecimal	Decimal
0x01	1

Instances (Single-Drive Mode)

The number of instances is fixed at three and is as shown below:

Instance	Description
0	Class
1	Host drive
2	22-COMM-E
3	22-SCM-232 or 22-HIM-* (when present)

Instances (Multi-Drive Mode)

The number of instances is fixed at one and is as shown below:

Instance	Description
0	Class
1	22-COMM-E

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Revision	UINT	1
2	Get	Max Instance	UINT	Total number of instances
6	Get	Max ID Number of Class Attributes	UINT	7
7	Get	Max ID Number of Instance Attributes	UINT	100

Identity Object *(Continued)*

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Vendor ID	UINT	1 = Allen-Bradley
2	Get	Device Type	UINT	127
3	Get	Product Code	UINT	Number identifying product name and rating
4	Get	Revision: Major Minor	STRUCT of: USINT USINT	 Value varies Value varies
5	Get	Status	WORD	Bit 0 = Owned Bit 2 = Configured Bit 10 = Recoverable fault Bit 11 = Unrecoverable fault
6	Get	Serial Number	UDINT	Unique 32-bit number
7	Get	Product Name	SHORT_ STRING	 Product name and rating
9	Get	Configuration Consistency Value	UINT	CRC or checksum representing the configuration of the product
100	Get	NVS Info	STRUCT of: UDINT SHORT_STRING	 First NVS instance Sub-assembly name

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x01	Yes	Yes	Get_Attributes_All
0x05	No	Yes	Reset
0x0E	Yes	Yes	Get_Attribute_Single

Assembly Object

Class Code

Hexadecimal	Decimal
0x04	4

Instances

Instance	Description
1	All I/O data being read from the DSI drives (read-only)
2	All I/O data written to the DSI drives (read/write)

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Revision	UINT	2
2	Get	Max Instance	UINT	2
100	Set	Control Timeout	UINT	Control timeout in seconds

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Number of Members	UINT	1
2	Get	Member List	ARRAY of STRUCT: UINT UINT Packed EPATH	 Size of member data Size of member path Member path
3	Conditional [1]	Data	Array of Bits	Data to be transferred
4	Get	Size	UINT	Size of assembly data in bits

[1] For instance 1, access rule for the data attribute is Get. For instance 2, it is Get/Set.

Important: Setting an Assembly object attribute can be done only when the Control Timeout (class attribute 100) has been set to a non-zero value.

Assembly Object *(Continued)*

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Register Object

Class Code

Hexadecimal	Decimal
0x07	7

Instances

Instance	Description	Input/ Output	Size (in bits)
1	Logic Command and Reference for all drives	Out	Varies (1)
2	Logic Status and Feedback for all drives	In	Varies (1)
3	Logic Command and Reference for Drive 0	Out	32
4	Logic Status and Feedback for Drive 0	In	32
5	Logic Command and Reference for Drive 1	Out	32
6	Logic Status and Feedback for Drive 1	In	32
7	Logic Command and Reference for Drive 2	Out	32
8	Logic Status and Feedback for Drive 2	In	32
9	Logic Command and Reference for Drive 3	Out	32
10	Logic Status and Feedback for Drive 3	In	32
11	Logic Command and Reference for Drive 4	Out	32
12	Logic Status and Feedback for Drive 4	In	32
13	Logic Command for all drives — mask-and-match register (2)	Out	Varies (1)
14	Logic Command for Drive 0 — mask-and-match register (2)	Out	32
15	Logic Command for Drive 1 — mask-and-match register (2)	Out	32
16	Logic Command for Drive 2 — mask-and-match register (2)	Out	32
17	Logic Command for Drive 3 — mask-and-match register (2)	Out	32
18	Logic Command for Drive 4 — mask-and-match register (2)	Out	32
19	Logic Command for Drive 0	Out	16
20	Logic Status for Drive 0	In	16
21	Reference for Drive 0	Out	16
22	Feedback for Drive 0	In	16
23	Logic Command for Drive 1	Out	16
24	Logic Status for Drive 1	In	16
25	Reference for Drive 1	Out	16
26	Feedback for Drive 1	In	16

(1) The size for this Register Object instance is 32 bits per drive. For example, if the adapter is operating in Multi-Drive mode, and is configured with Drives 0 through 4, then the size is 160 bits.

(2) The structure for this Register Object instance is a Mask word followed by a Command word for each drive. The Logic Command for each drive is set to the value of the second word of the data where there are ones in the first word of the data.

Logic Command = (Logic Command and not Mask word) or (Command word and Mask word)

Register Object *(Continued)*

Instances (Continued)

Instance	Description	Input/ Output	Size (in bits)
27	Logic Command for Drive 2	Out	16
28	Logic Status for Drive 2	In	16
29	Reference for Drive 2	Out	16
30	Feedback for Drive 2	In	16
31	Logic Command for Drive 3	Out	16
32	Logic Status for Drive 3	In	16
33	Reference for Drive 3	Out	16
34	Feedback for Drive 3	In	16
35	Logic Command for Drive 4	Out	16
36	Logic Status for Drive 4	In	16
37	Reference for Drive 4	Out	16
38	Feedback for Drive 4	In	16

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Revision	UINT	1
2	Get	Max Instance	UINT	38
3	Get	Number of Instances	UINT	38
100	Set	Control Timeout	UINT	Control timeout in seconds

Register Object *(Continued)*

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Bad Flag	BOOL	If set to 1, then attribute 4 contains invalid, bad or otherwise corrupt data. 0 = good 1 = bad
2	Get	Direction	BOOL	Direction of data transfer 0 = Input (Drive to EtherNet/IP) 1 = Output (EtherNet/IP to Drive)
3	Get	Size	UINT	Size of register data in bits
4	Conditional (1)	Data	ARRAY of BITS	Data to be transferred

(1) The access rule of Set is optional if attribute 2, Direction = 1. If Direction = 0, the access rule is Get.

Important: Setting a Register object attribute can be done only when the Control Timeout (class attribute 100) has been set to a non-zero value.

	Implemented for:		
Service Code	Class	Instance	Service Name
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Parameter Object

Class Code

Hexadecimal	Decimal
0x0F	15

Instances (Single-Drive Mode)

The number of instances is as shown below:

Instance	Description
0	Class
1	Drive Parameter 1
↕	↕
n	Drive Parameter n (1)
n + 1	Adapter Parameter 1
↕	↕
n + m	Adapter Parameter m (2)

(1) n represents the number of parameters in the drive.

(2) m represents the number of parameters in the adapter.

Instances (Multi-Drive Mode)

The number of instances is as shown below:

Instance	Description
0	Class
1	Adapter Parameter 1
↕	↕
m	Adapter Parameter m (1)

(1) m represents the number of parameters in the adapter.

In addition, the parameters for the other DSI devices can be accessed using the instance-offset encoding shown in the table below:

Instances (Dec.)	Single-Drive Mode	Multi-Drive Mode
16384 – 17407	Instances 0 – 1023 in the adapter	Instances 0 – 1023 in the adapter
17408 – 18431	Instances 0 – 1023 in the drive	Instances 0 – 1023 in Drive 0
18432 – 19455	Not supported	Instances 0 – 1023 in Drive 1
19456 – 20479	Not supported	Instances 0 – 1023 in Drive 2
20480 – 21503	Not supported	Instances 0 – 1023 in Drive 3
21504 – 22527	Not supported	Instances 0 – 1023 in Drive 4

Parameter Object *(Continued)*

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Revision	UINT	1
2	Get	Max Instance	UINT	Number of parameters
8	Get	Parameter Class Descriptor	WORD	0 = False, 1 = True Bit 0 = Supports parameter instances Bit 1 = Supports full attributes Bit 2 = Must do NVS save command Bit 3 = Parameters are stored in NVS
9	Get	Configuration Assembly Instance	UINT	0
10	Get	Native Language	USINT	0 = English 1 = French 2 = Spanish 3 = Italian 4 = German 5 = Japanese 6 = Portuguese 7 = Mandarin Chinese 8 = Russian 9 = Dutch

Parameter Object *(Continued)*

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	(1)	Parameter Value	(2)	(3)
2	Get	Link Path Size	USINT	0 = No link specified n = The size of Attribute 3 in bytes
3	Get	Link Path		(4)
4	Get	Descriptor	WORD	0 = False, 1 = True Bit 1 = Supports ENUMs Bit 2 = Supports scaling Bit 3 = Supports scaling links Bit 4 = Read only Bit 5 = Monitor Bit 6 = Extended precision scaling
5	Get	Data Type	USINT	0xC2 = SINT (8-bits) 0xC3 = INT (16-bits) 0xC4 = DINT (32-bits) 0xC6 = USINT (8-bits) 0xC7 = UINT (16-bits) 0xCA = REAL (32-bits) 0xD2 = WORD (16-bits)
6	Get	Data Size	USINT	(3)
7	Get	Parameter Name String	SHORT_ STRING	(3)
8	Get	Units String	SHORT_ STRING	(3)
9	Get	Help String	SHORT_ STRING	Null string
10	Get	Minimum Value	(1)	(3)
11	Get	Maximum Value	(1)	(3)
12	Get	Default Value	(1)	(3)
13	Get	Scaling Multiplier	UINT	(3)
14	Get	Scaling Divisor	UINT	(3)
15	Get	Scaling Base	UINT	(3)
16	Get	Scaling Offset	UINT	(3)
17	Get	Multiplier Link	UINT	(3)
18	Get	Divisor Link	UINT	(3)
19	Get	Base Link	UINT	(3)
20	Get	Offset Link	UINT	(3)
21	Get	Decimal Precision	USINT	(3)

(1) Access rule is defined in bit 4 of instance attribute 4. 0 = Get/Set, 1 = Get.

(2) Specified in descriptor, data type, and data size.

(3) Value varies based on parameter instance.

(4) Refer to the CIP Common specification for a description of the link path.

Parameter Object *(Continued)*

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x01	Yes	Yes	Get_Attribute_All
0x05	Yes	No	Reset
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single
0x4B	No	Yes	Get_Enum_String

Parameter Group Object

Class Code

Hexadecimal	Decimal
0x10	16

Instances

The number of instances depends on the number of groups in the device. A group of adapter parameters is appended to the list of groups in the device. The total number of groups can be read in Instance 0, Attribute 2.

Number	Description
0	Class Attributes
1	Drive Group 1 Attributes
⋮	⋮
n	Drive Group n Attributes (1)
n + 1	Adapter Group Attributes

(1) n represents the number of parameter groups in the drive.

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Parameter group version	UINT	1
2	Get	Max Instance	UINT	Total number of groups
8	Set	Native Language	USINT	0 = English 1 = French 2 = Spanish (Mexican) 3 = Italian 4 = German 5 = Japanese 6 = Portuguese 7 = Mandarin Chinese 8 = Russian 9 = Dutch

Parameter Group Object *(Continued)*

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Group Name String	SHORT_ STRING	Group name
2	Get	Number of Members in Group	UINT	Number of parameters in group.
3	Get	1st Parameter Number in Group	UINT	(1)
4	Get	2nd Parameter Number in Group	UINT	(1)
n	Get	⋮	UINT	(1)

(1) Value varies based on group instance.

Services

	Implemented for:		
Service Code	Class	Instance	Service Name
0x0E	Yes	Yes	Get_Attribute_Single
0x01	Yes	Yes	Get_Attributes_All

PCCC Object

Class Code

Hexadecimal	Decimal
0x67	103

Instances

Supports Instance 1.

Class Attributes

Not supported.

Instance Attributes

Not supported.

Services

	Implemented for:		
Service Code	Class	Instance	Service Name
0x4B	No	Yes	Execute_PCCC
0x4C	No	Yes	Execute_DH+

Message Structure for Execute_PCCC

Request		
Name	Data Type	Description
Length	USINT	Length of requestor ID
Vendor	UINT	Vendor number of requestor
Serial Number	UDINT	ASA serial number of requestor
Other	Product Specific	Identifier of user, task, etc. on the requestor
CMD	USINT	Command byte

Response		
Name	Data Type	Description
Length	USINT	Length of requestor ID
Vendor	UINT	Vendor number of requestor
Serial Number	UDINT	ASA serial number of requestor
Other	Product Specific	Identifier of user, task, etc. on the requestor
CMD	USINT	Command byte

PCCC Object *(Continued)*

Message Structure for Execute_PCCC (Continued)

Request		
Name	**Data Type**	**Description**
STS	USINT	0
TNSW	UINT	Transport word
FNC	USINT	Function code. Not used for all CMD's.
PCCC_ params	ARRAY of USINT	CMD/FNC specific parameters

Response		
Name	**Data Type**	**Description**
STS	USINT	Status byte
TNSW	UINT	Transport word. Same value as the request.
EXT_STS	USINT	Extended status. Not used for all CMD's.
PCCC_ results	ARRAY of USINT	CMD/FNC specific result data

Message Structure for Execute_DH+

Request		
Name	**Data Type**	**Description**
DLink	UINT	Destination Link ID
DSta	USINT	Destination Station number
DUser	USINT	Destination "User" number
SLink	UINT	Source Link ID
SSta	USINT	Source Station number
SUser	USINT	Source User number
CMD	USINT	Command byte
STS	USINT	0
TNSW	UINT	Transport word
FNC	USINT	Function code; not used for all CMD's
PCCC_ params	ARRAY of USINT	CMD/FNC specific parameters

Response		
Name	**Data Type**	**Description**
DLink	UINT	Destination Link ID
DSta	USINT	Destination Station number
DUser	USINT	Destination "User" number
SLink	UINT	Source Link ID
SSta	USINT	Source Station number
SUser	USINT	Source User number
CMD	USINT	Command byte
STS	USINT	Status byte
TNSW	UINT	Transport word. Same value as the request.
EXT_STS	USINT	Extended Status; not used for all CMD's
PCCC_ results	ARRAY of USINT	CMD/FNC specific result data

PCCC Object *(Continued)*

The adapter supports the following PCCC command types:

CMD	FNC	Description
0x06	0x03	Identify host and some status
0x0F	0x67	PLC-5 typed write
0x0F	0x68	PLC-5 typed read
0x0F	0x95	Encapsulate other protocol
0x0F	0xA2	SLC 500 protected typed read with 3 address fields
0x0F	0xAA	SLC 500 protected typed write with 3 address fields
0x0F	0xA1	SLC 500 protected typed read with 2 address fields
0x0F	0xA9	SLC 500 protected typed write with 2 address fields
0x0F	0x00	Word range read
0x0F	0x01	Word range write

See DF1 Protocol and Command Set Manual, Allen-Bradley Publication No. 1770-6.5.16.

N-Files

<table>
<tr><th>N-File</th><th colspan="4">Description</th></tr>
<tr><td>N40</td><td colspan="4">This N-file lets you use Emulated Block Transfer messages to read and write many types of DPI messages. To use Emulated Block Transfer messages, you send a Write message to N40:0 – N40:63, wait until the adapter responds with a reply message, and then read the response data in N40:0 – N40:63 with a Read message.

For details about Block Transfer messages and the data required for each byte in the N-File, refer to the Remote I/O Adapter User Manual, Publication 20COMM-UM004….

Bits 15 to 8 are the Most Significant Byte. Bits 7 to 0 are the Least Significant Byte.</td></tr>
<tr><td></td><td colspan="2">Write</td><td colspan="2">Read</td></tr>
<tr><td>Bits</td><td>15 0</td><td></td><td>15 0</td><td></td></tr>
<tr><td>N40:0</td><td>0x00</td><td>Length (in Bytes)</td><td>0x00</td><td>Length (in Bytes)</td></tr>
<tr><td>N40:1</td><td>DPI Port (1)</td><td>0x81</td><td>Status Size</td><td>Status Type</td></tr>
<tr><td>N40:2</td><td>0x00</td><td>CIP Service</td><td colspan="2" rowspan="5">Data
(length varies based on message)</td></tr>
<tr><td>N40:3</td><td colspan="2">CIP Class</td></tr>
<tr><td>N40:4</td><td colspan="2">CIP Instance</td></tr>
<tr><td>N40:5</td><td colspan="2">CIP Attribute</td></tr>
<tr><td>N40:6
⋮
N40:63</td><td colspan="2">Data
(length varies based on message)</td></tr>
</table>

(1) Use the following DPI Port Assignment table to determine the value for the DPI port.

PCCC Object *(Continued)*

N-Files (Continued)

DPI Port Assignments

DPI Port No.	Single-Drive Mode	Multi-Drive Mode
0	The drive	Drive 0
1	The adapter	Drive 1
2	The slave	Drive 2
3	Not supported	Drive 3
4	Not supported	Drive 4
5	Not supported	The adapter

N-File	Description	
	For Single-Drive Mode Only	
N41	This N-file lets you read and write control I/O messages. You can write control I/O messages only when all of the following conditions are true: • The adapter is not receiving I/O from a scanner. For example, there is no scanner on the network, the scanner is in idle (program) mode, the scanner is faulted, or the adapter is not mapped to the scanner. • The value of N42:3 is set to a non-zero value.	
	Write	*Read*
N41:0	Logic Command Word	Logic Status Word
N41:1	Unused	Unused
N41:2	Reference	Feedback
N42	This N-file lets you read and write some values configuring the port	
N42:3	Time-out (read/write): Time (in seconds) allowed between messages to the N41 or N44 file. If the adapter does not receive a message in the specified time, it performs the fault action configured in its [Comm Flt Action] parameter.	
N42:7	Adapter Port Number (read only): DPI port on the drive to which the adapter is connected.	
N42:8	Peer Adapters (read only): Bit field of devices having DPI Peer capabilities.	

PCCC Object *(Continued)*

N-Files (Continued)

N-File	Description	
	For Multi-Drive Mode Only	
N44	This N-file lets you read and write control I/O messages. You can write control I/O messages only when all of the following conditions are true: • The adapter is not receiving I/O from a scanner. For example, there is no scanner on the network, the scanner is in idle (program) mode, the scanner is faulted, or the adapter is not mapped to the scanner. • The value of N42:3 is set to a non-zero value.	
	Write	*Read*
N44:0	Drive 0 Logic Command	Drive 0 Logic Status
N44:1	Unused	Unused
N44:2	Drive 0 Reference	Drive 0 Feedback
N44:3	Drive 1 Logic Command	Drive 1 Logic Status
N44:4	Drive 1 Reference	Drive 1 Feedback
N44:5	Drive 2 Logic Command	Drive 2 Logic Status
N44:6	Drive 2 Reference	Drive 2 Feedback
N44:7	Drive 3 Logic Command	Drive 3 Logic Status
N44:8	Drive 3 Reference	Drive 3 Feedback
N44:9	Drive 4 Logic Command	Drive 4 Logic Status
N44:10	Drive 4 Reference	Drive 4 Feedback

PCCC Object *(Continued)*

N-Files (Continued)

Important: If your controller or HMI platform supports CIP messaging, use the CIP Parameter object to get and set parameters.

N-File	Description	
N10 – N18	These N-files let you read and write parameter values in the drive and the adapter.	
	Single-Drive Mode	*Multi-Drive Mode*
N10:0 N10:1 – 999	Number of parameters in the drive Drive parameters 1 – 999	Number of parameters in Drive 0 Drive 0 parameters 1 - 999
N11:0 – 999	Drive parameters 1000 – 1999	Drive 0 parameters 1000 - 1999
N12:0 – 999	Drive parameters 2000 – 2999	Drive 0 parameters 2000 - 2999
N13:0	Number of parameters in this adapter	Number of parameters in this adapter
N13:1 – 999	Parameters 1 – 999 in this adapter	Parameters 1 – 999 in this adapter
N14:0	Number of parameters in this adapter	Number of parameters in Drive 1
N14:1 – 999	Parameters 1 – 999 in this adapter	Drive 1 parameters 1 – 999
N15:0	Number of parameters in the slave	Number of parameters in Drive 2
N15:1 – 999	Parameters 1 – 999 in the slave	Drive 2 parameters 1 – 999
N16:0	Not supported	Number of parameters in Drive 3
N16:1 – 999	Not supported	Drive 3 parameters 1 – 999
N17:0	Not supported	Number of parameters in Drive 4
N17:1 – 999	Not supported	Drive 4 parameters 1 – 999
N18:0	Not supported	Number of parameters in this adapter
N18:1 – 999	Not supported	Parameters 1 – 999 in this adapter

DPI Device Object

Class Code

Hexadecimal	Decimal
0x92	146

Instances

The number of instances depends on the number of components in the device. The total number of components can be read in Instance 0, Class Attribute 4.

Instances (Dec.)	Single-Drive Mode	Multi-Drive Mode
0 – 16383	Instances 0 – 16383 in the drive	Instances 0 – 16383 in Drive 0
16384 – 17407	Instances 0 – 1023 in the adapter	Instances 0 – 1023 in the adapter
17408 – 18431	Instances 0 – 1023 in the adapter	Instances 0 – 1023 in Drive 1
18432 – 19455	Instances 0 – 1023 in the slave	Instances 0 – 1023 in Drive 2
19456 – 20479	Not supported	Instances 0 – 1023 in Drive 3
20480 – 21503	Not supported	Instances 0 – 1023 in Drive 4
21504 – 22527	Not supported	Instances 0 – 1023 in the adapter

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Family Code	BYTE	Code identifying the device.
1	Get	Family Text	STRING[16]	Text identifying the device.
2	Set	Language Code	BYTE	0 = English 1 = French 2 = Spanish 3 = Italian 4 = German 5 = Japanese 6 = Portuguese 7 = Mandarin Chinese 8 = Russian 9 = Dutch
3	Get	Product Series	BYTE	1 = A 2 = B …
4	Get	Number of Components	BYTE	Number of components (e.g., main control board, I/O boards) in the device.

DPI Device Object *(Continued)*

Class Attributes (Continued)

Attribute ID	Access Rule	Name	Data Type	Description
5	Set	User Definable Text	STRING[16]	Text identifying the device with a user-supplied name
6	Get	Status Text	STRING[12]	Text describing the status of the device.
7	Get	Configuration Code	BYTE	Identification of variations.
8	Get	Configuration Text	STRING[16]	Text identifying a variation of a family device.
9	Get	Brand Code	WORD	0x0001 = Allen-Bradley
11	Get	NVS Checksum	WORD	Checksum of the Non-Volatile Storage in a device.
12	Get	Class Revision	WORD	2 = DPI
13	Get	Character Set Code	BYTE	0 = SCANport HIM 1 = ISO 8859-1 (Latin 1) 2 = ISO 8859-2 (Latin 2) 3 = ISO 8859-3 (Latin 3) 4 = ISO 8859-4 (Latin 4) 5 = ISO 8859-5 (Cyrillic) 6 = ISO 8859-6 (Arabic) 7 = ISO 8859-7 (Greek) 8 = ISO 8859-8 (Hebrew) 9 = ISO 8859-9 (Turkish) 10 = ISO 8859-10 (Nordic) 255 = ISO 10646 (Unicode)
15	Get	Languages Supported	STRUCT of: BYTE BYTE[n]	 Number of Languages Language Codes (See Class Attribute 2)
16	Get	Date of Manufacture	STRUCT of: WORD BYTE BYTE	 Year Month Day
17	Get	Product Revision	STRUCT of: BYTE BYTE	 Major Firmware Release Minor Firmware Release
18	Get	Serial Number	DWORD	Value between 0x00 and 0xFFFFFFFF

DPI Device Object *(Continued)*

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
3	Get	Component Name	STRING[32]	Name of the component
4	Get	Component Firmware Revision	STRUCT of: BYTE BYTE	 Major Revision Minor Revision
5	Get	Component Hardware Change Number	BYTE	
8	Get	Component Serial Number	DWORD	Value between 0x00 and 0xFFFFFFFF

Services

	Implemented for:		
Service Code	Class	Instance	Service Name
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

DPI Parameter Object

Class Code

Hexadecimal	Decimal
0x93	147

Instances

The number of instances depends on the number of parameters in the device. The total number of parameters can be read in Instance 0, Attribute 0.

Instances (Dec.)	Single-Drive Mode	Multi-Drive Mode
0 – 16383	Instances 0 – 16383 in the drive	Instances 0 – 16383 in Drive 0
16384 – 17407	Instances 0 – 1023 in the adapter	Instances 0 – 1023 in the adapter
17408 – 18431	Instances 0 – 1023 in the adapter	Instances 0 – 1023 in Drive 1
18432 – 19455	Instances 0 – 1023 in the slave	Instances 0 – 1023 in Drive 2
19456 – 20479	Not supported	Instances 0 – 1023 in Drive 3
20480 – 21503	Not supported	Instances 0 – 1023 in Drive 4
21504 – 22527	Not supported	Instances 0 – 1023 in the adapter

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Number of Instances	WORD	Number of parameters in the device
1	Set	Write Protect Password	WORD	0 = Password disabled n = Password
2	Set	NVS Command Write	BYTE	0 = No Operation 1 = Store values in active memory to NVS 2 = Load values in NVS to active memory 3 = Load default values to active memory
3	Get	NVS Parameter Value Checksum	WORD	Checksum of all parameter values in a user set in NVS
4	Get	NVS Link Value Checksum	WORD	Checksum of parameter links in a user set in NVS
5	Get	First Accessible Parameter	WORD	First parameter available if parameters are protected by passwords. A “0” indicates all parameters are protected.
7	Get	Class Revision	WORD	2 = DPI
8	Get	First Parameter Processing Error	WORD	The first parameter that has been written with a value outside of its range. A “0” indicates no errors.
9	Set	Link Command	BYTE	0 = No Operation 1 = Clear All Parameter Links (This does not clear links to function blocks.)

DPI Parameter Object *(Continued)*

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
7	Get	DPI Online Read Full	STRUCT of: BOOL[32] CONTAINER(1) CONTAINER CONTAINER CONTAINER WORD WORD STRING[4] UINT UINT UINT INT BYTE[3] BYTE STRING[16]	 Descriptor (Refer to pages C-26 – C-27) Parameter value Minimum value Maximum value Default value Next parameter Previous parameter Units (e.g., Amp, Hz) Multiplier (2) Divisor (2) Base (2) Offset (2) Link (source of the value) (0 = no link) Always zero (0) Parameter name
8	Get	DPI Descriptor	BOOL[32]	Descriptor (Refer to pages C-26 – C-27)
9	Get/Set	DPI Parameter Value	Various	Parameter value in NVS. (3)
10	Get/Set	DPI RAM Parameter Value	Various	Parameter value in temporary memory.
11	Get/Set	DPI Link	BYTE[3]	Link (parameter or function block that is the source of the value) (0 = no link)
12	Get	Help Object Instance	WORD	ID for help text for this parameter
13	Get	DPI Read Basic	STRUCT of: BOOL[32] CONTAINER CONTAINER CONTAINER CONTAINER STRING[16] STRING[4]	 Descriptor (Refer to pages C-26 – C-27) Parameter value Minimum value Maximum value Default value Parameter name Units (e.g., Amp, Hz)
14	Get	DPI Parameter Name	STRING[16]	Parameter name
15	Get	DPI Parameter Alias	STRING[16]	Customer supplied parameter name. Only supported by PowerFlex 700S at time of publication.
16	Get	Parameter Processing Error	BYTE	0 = No error 1 = Value is less than the minimum 2 = Value is greater than the maximum

(1) A CONTAINER is a 32-bit block of data that contains the data type used by a parameter value. If signed, the value is sign extended. Padding is used in the CONTAINER to ensure that it is always 32-bits.

(2) This value is used in the formulas used to convert the parameter value between display units and internal units. Refer to Formulas for Converting on page C-27.

(3) Do NOT continually write parameter data to NVS. Refer to the attention on page 6-1.

DPI Parameter Object *(Continued)*

Descriptor Attributes

Bit	Name	Description
0	Data Type (Bit 1)	Right bit is least significant bit (0). 000 = BYTE used as an array of Boolean 001 = WORD used as an array of Boolean 010 = BYTE (8-bit integer) 011 = WORD (16-bit integer) 100 = DWORD (32-bit integer) 101 = TCHAR (8-bit (not unicode) or 16-bits (unicode)) 110 = REAL (32-bit floating point value) 111 = Use bits 16, 17, 18
1	Data Type (Bit 2)	
2	Data Type (Bit 3)	
3	Sign Type	0 = unsigned 1 = signed
4	Hidden	0 = visible 1 = hidden
5	Not a Link Sink	0 = Parameter can sink a link 1 = Parameter cannot sink a link
6	Not Recallable	0 = Recallable from NVS 1 = Not Recallable from NVS
7	ENUM	0 = No ENUM text 1 = ENUM text
8	Writable	0 = Read only 1 = Read/write
9	Not Writable When Enabled	0 = Writable when enabled (e.g., drive running) 1 = Not writable when enabled
10	Instance	0 = Parameter value is not a Reference to another parameter 1 = Parameter value refers to another parameter
11	Reserved	Must be zero
12	Decimal Place (Bit 0)	Number of digits to the right of the decimal point. 0000 = 0 1111 = 15
13	Decimal Place (Bit 1)	
14	Decimal Place (Bit 2)	
15	Decimal Place (Bit 3)	
16	Extended Data Type (Bit 1)	Right bit is least significant bit (16). 000 = Reserved 001 = DWORD used as an array of Boolean 010 = Reserved 011 = Reserved 100 = Reserved 101 = Reserved 110 = Reserved 111 = Reserved
17	Extended Data Type (Bit 2)	
18	Extended Data Type (Bit 2)	

DPI Parameter Object *(Continued)*

Descriptor Attributes (Continued)

Bit	Name	Description
19	Parameter Exists	Reserved
20	Not Used	Reserved
21	Formula Links	Reserved
22	Access Level (Bit 1)	Reserved
23	Access Level (Bit 2)	Reserved
24	Access Level (Bit 3)	Reserved
25	Writable ENUM	Reserved
26	Not a Link Source	0 = Parameter can be a source for a link 1 = Parameter cannot be a source for a link
27	Enhanced Bit ENUM	Reserved
28	Enhanced ENUM	Reserved
29	Not Used	Reserved
30	Not Used	Reserved
31	Not Used	Reserved

Formulas for Converting

Display Value = ((Internal Value + Offset) x Multiplier x Base) / (Divisor x $10^{\text{Decimal Places}}$)

Internal Value = ((Display Value x Divisor x $10^{\text{Decimal Places}}$) / (Multiplier x Base)) - Offset

Common Services

	Implemented for:		
Service Code	Class	Instance	Service Name
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Object Specific Services

Service Code	Service Name
0x32	Get_Attributes_Scattered (1)
0x34	Set_Attributes_Scattered (1)

(1) The instance and attribute are ignored for these services.

The table below lists the parameters for the Get_Attributes_Scattered and Set_Attributes_Scattered object-specific service:

Name	Data Type	Description
Scattered Parameters	STRUCT of	—
Parameter Number	WORD	Parameter to read or write
Parameter Value	WORD	Parameter value to read or write (zero when reading)

Important: The STRUCT may repeat up to 64 times in a single message.

DPI Fault Object

Class Code

Hexadecimal	Decimal
0x97	151

Products such as PowerFlex drives use this object for faults. Adapters use this object for events.

Instances

The number of instances depends on the maximum number of faults or events supported in the queue. The maximum number of faults/events can be read in Instance 0, Attribute 2.

Instances (Dec.)	Single-Drive Mode	Multi-Drive Mode
0 – 16383	Instances 0 – 16383 in the drive	Instances 0 – 16383 in Drive 0
16384 – 17407	Instances 0 – 1023 in the adapter	Instances 0 – 1023 in the adapter
17408 – 18431	Instances 0 – 1023 in the adapter	Instances 0 – 1023 in Drive 1
18432 – 19455	Instances 0 – 1023 in the slave	Instances 0 – 1023 in Drive 2
19456 – 20479	Not supported	Instances 0 – 1023 in Drive 3
20480 – 21503	Not supported	Instances 0 – 1023 in Drive 4
21504 – 22527	Not supported	Instances 0 – 1023 in the adapter

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Class Revision	WORD	Revision of object
2	Get	Number of Instances	WORD	Maximum number of faults/events that the device can record in its queue
3	Set	Fault Command Write	BYTE	0 = No Operation 1 = Clear Fault/Event 2 = Clear Fault/Event Queue 3 = Reset Device
4	Get	Fault Trip Instance Read	WORD	Fault that tripped the device. For adapters, this value is always 1 when faulted.
5	Get	Fault Data List	STRUCT of: BYTE BYTE WORD[n]	Reserved
6	Get	Number of Recorded Faults	WORD	Number of faults/events in the queue. A “0” indicates the fault queue is empty.
7	Get	Fault Parameter Reference	WORD	Reserved

DPI Fault Object *(Continued)*

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description

0	Get	Full/All Information	STRUCT of:
			  WORD
			STRUCT of:
			  BYTE
			  BYTE
			  STRING[16]
			STRUCT of:
			  LWORD
			  BOOL[16]
			
			WORD
			CONTAINER[n]
				
				Fault code
				Fault source
				DPI port
				DPI Device Object
				Fault text
				Fault time stamp
				Timer value (0 = Timer not supported)
				BOOL[0]: (0 = invalid data, 1 = valid data)
				BOOL[1]: (0 = elapsed time, 1 = real time)
				BOOL[2 - 15]: Not used
				Reserved
				Reserved

1	Get	Basic Information	STRUCT of:
			  WORD
			STRUCT of:
			  BYTE
			  BYTE
			STRUCT of:
			  LWORD
			  BOOL[16]
				
				Fault code
				Fault source
				DPI port
				DPI Device Object
				Fault time stamp
				Timer value (0 = Timer not supported)
				BOOL[0]: (0 = invalid data, 1 = valid data)
				BOOL[1]: (0 = elapsed time, 1 = real time)
				BOOL[2 - 15]: Not used

Services

	Implemented for:		
Service Code	Class	Instance	Service Name
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

DPI Diagnostic Object

Class Code

Hexadecimal	Decimal
0x99	153

Instances

The number of instances depends on the maximum number of diagnostic items in the device. The total number of diagnostic items can be read in Instance 0, Attribute 2.

Instances (Dec.)	Single-Drive Mode	Multi-Drive Mode
0 – 16383	Instances 0 – 16383 in the drive	Instances 0 – 16383 in Drive 0
16384 – 17407	Instances 0 – 1023 in the adapter	Instances 0 – 1023 in the adapter
17408 – 18431	Instances 0 – 1023 in the adapter	Instances 0 – 1023 in Drive 1
18432 – 19455	Instances 0 – 1023 in the slave	Instances 0 – 1023 in Drive 2
19456 – 20479	Not supported	Instances 0 – 1023 in Drive 3
20480 – 21503	Not supported	Instances 0 – 1023 in Drive 4
21504 – 22527	Not supported	Instances 0 – 1023 in the adapter

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Class Revision	WORD	1
2	Get	Number of Instances	WORD	Number of diagnostic items in the device
3	Get	ENUM Offset	WORD	DPI ENUM object instance offset

DPI Diagnostic Object *(Continued)*

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Full/All Info	STRUCT of: BOOL[32] CONTAINER (1) CONTAINER CONTAINER CONTAINER WORD WORD STRING[4] UINT UINT UINT INT DWORD STRING[16]	 Descriptor (Refer to pages C-26 – C-27) Value Minimum value Maximum value Default value Pad Word Pad Word Units (e.g., Amp, Hz) Multiplier (2) Divisor (2) Base (2) Offset (2) Link (source of the value) (0 = no link) Always zero (0) Parameter name
1	Get/Set	Value	Various	Diagnostic item value

(1) A CONTAINER is a 32-bit block of data that contains the data type used by a value. If signed, the value is sign extended. Padding is used in the CONTAINER to ensure that it is always 32-bits.

(2) This value is used in the formulas used to convert the value between display units and internal units. Refer to Formulas for Converting on page C-27.

Services

Service Code	Implemented for: Class	Implemented for: Instance	Service Name
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

TCP/IP Interface Object

Class Code

Hexadecimal	Decimal
0xF5	245

Instances

The adapter supports one instance of the TCP/IP Interface object.

Number	Description
0	Class Attributes
1	Object Attributes

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Revision	UINT	The revision of this object

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Status of TCP/IP Network Interface	DWORD	0 = Not configured 1 = Valid configuration 2 to 15 = Reserved
2	Get	Configuration Capability	DWORD	Bit	Value (0 = False, 1 = True) 0 = Supports BOOTP 1 = DNS Client (Able to resolve host names by query to DNS server) 2 = DHCP Client (Able to obtain network configuration through DHCP) 3 = DHCP-DNS Update (Able to send its host name in the DHCP request) 4 = Configuration Settable (Able to set the network configuration via TCP/IP) 5 to 31 = Reserved

TCP/IP Interface Object *(Continued)*

Instance Attributes (Continued)

Attribute ID	Access Rule	Name	Data Type	Description
3	Set	Configuration Control	DWORD	Bit I (Value) 1 – 3 = Startup configuration (0 = Use configuration saved in NVS) (1 = Obtain configuration via BOOTP) (2 = Obtain configuration via DHCP) (3 to 15 = Reserved) 4 = DNS Enabled (Resolves host names by query to DNS server) 5 to 31 = Reserved
4	Get	Physical Link Object	STRUCT of: UINT Padded EPATH	 Path size Path
5	Get	Interface Configuration	STRUCT of UDINT UDINT UDINT UDINT UDINT STRING	 Adapter's IP address Adapter's subnet mask Adapter's gateway address Primary name server Secondary name server Default domain name
6	Get	Host Name	STRING	Host name when using DHCP

Services

	Implemented for:		
Service Code	Class	Instance	Service Name
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single

Ethernet Link Object

Class Code

Hexadecimal	Decimal
0xF6	246

Instances

The adapter supports one instance of the TCP/IP Interface object.

Number	Description
0	Class Attributes
1	Object Attributes

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Revision	UINT	The revision of this object

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Interface Speed	UDINT	Speed in megabits per second (Mbps)
2	Get	Interface Flags	DWORD	Bit I (Value) 0 = Link status (0 = inactive, 1 = active) 1 = Duplex (0 = half duplex, 1 = full duplex) 2 to 31 = Reserved
3	Get	Physical Address	USINT[6]	MAC address (XX-XX-XX-XX-XX-XX) The first octet (USINT[0]) is on the left.

Ethernet Link Object *(Continued)*

Instance Attributes (Continued)

Attribute ID	Access Rule	Name	Data Type	Description
4	Get	Interface Counters	STRUCT of:	
			UDINT	Octets received
			UDINT	Unicast packets received
			UDINT	Non-unicast packets received
			UDINT	Inbound packets received but discarded
			UDINT	Inbound packets with errors (not discarded)
			UDINT	Inbound packets with unknown protocol
			UDINT	Octets sent
			UDINT	Unicast packets sent
			UDINT	Non-unicast packets sent
			UDINT	Outbound packets discarded
			UDINT	Outbound packets with errors
5	Get	Media Counters	STRUCT of:	RX = Received, TX = Transmitted
			UDINT	RX frames not having integral number of octets long
			UDINT	RX frames not passing FCS check
			UDINT	TX frames having one collision
			UDINT	TX frames having multiple collisions
			UDINT	Number of times of SQE test error message
			UDINT	TX Frames delayed first attempt by busy medium
			UDINT	Collisions detected later than 512 bit-times in trans.
			UDINT	TX frames failing due to excessive collisions
			UDINT	TX frames failing due to intern MAC sublayer TX error
			UDINT	Times of carrier sense condition loss during trans.
			UDINT	RX frames exceeding the maximum frame size
			UDINT	RX frames failing due to intern MAC sublayer RX error

Services

	Implemented for:		
Service Code	Class	Instance	Service Name
0x0E	Yes	Yes	Get_Attribute_Single
0x4C	No	Yes	Get_and_Clear

Notes: